Biological Data Transformation in Pathway Simulation*

Abel Gómez,[1] Artur Boronat,[1] Jose Á. Carsí,[1] and Isidro Ramos[1]

[1] Departament de Sistemes Informàtics i Computació. Universitat Politècnica de València. Camino de Vera, s/n. 46022 València. España.

This work shows how Model-Driven Software Development (MDSD) can be applied in the bioinformatics field since biological data structures can be easily expressed by means of models. The existence of several heterogeneous data sources is usual in the bioinformatics context. In order to validate the information stored in these data sources, several formalisms and simulation tools have been adopted. The process of importing data from the source databases and introducing it in the simulation tools is usually done by hand. This work describes how to overcome this drawback by applying MDSD techniques (e.g. model transformations). Such techniques allow us to automate the data migration process between source databases and simulation tools, making the transformation process independent of the data persistence format, obtaining more modular tools and generating traceability information automatically.

MOTIVATION

The traditional sequence of "experiment → analysis → publication" is changing to "experiment → data organization → analysis → publication" [6]. This is because, nowadays, data is not only obtained from experiments, but also from simulations. The great amount of new data that can be generated from these experiments is not always homogeneous and may be stored in different databases.

This scenario is found especially when analyzing and simulating cell-signaling mechanisms (*Signal Transduction Pathways*). In studies of this type, it is very common to find both independent databases and modeling tools. Thus, the data of the databases must be converted from the source databases to the simulation tools in order to be used.

Model-Driven Software Development (MDSD) is an approach that attempts to solve problems of this kind. In MDSD, a model is a data structure that can be defined by means of a modeling language (usually called *metamodel*). Using models in a MDSD process allows the automation of the development and evolution of the software applications thanks to generative programming techniques [5] such as model transformations and code generation.

Dealing with data from the MDSD perspective helps to develop tools where the data processing mechanisms are independent of the final persistence format, obtaining more modular tools. This also helps to automate the data migration process by means of model transformation techniques. All these factors reduce the costs of the software development process, directly increasing the productivity of the users/biologists.

BACKGROUND

In organisms, proteins have a wide variety of functions and they interact with each other in similar multifaceted ways. These interactions of proteins are described by means of signal transduction pathways or networks, which are typically represented as certain kinds of maps.

* This work has been supported by the Spanish Government under the National Program for Research, Development and Innovation META TIN2006-15175-C05-01 and the *Generalitat Valenciana* ref. ACOMP07/216

These signal transduction pathways are composed by experts, who study the relevant literature that is produced by various groups worldwide doing research on very small parts of signal transduction pathways in different kind of organisms. This information is then composed bottom-up to a signal transduction pathway and introduced into databases to provide an integrated view on the entire pathway.

Examples for such signal transduction pathway databases are TRANSPATH® [11], KEGG [10] or Reactome [9]. They usually provide a web interface for interactive searches and also make their data available as text files in flat file or XML format.

Understanding the flow of information inside a cell is fundamental for an in-depth understanding of the functioning of a cell as a whole. Therefore, modeling and simulating this information flow is beneficial because it helps to understand the flow of signals in a complex network, to test hypotheses in silico before validating them with experiments, and to validate the data collected about a certain signal transduction pathway.

We are currently working on one of the major signal transduction pathways databases, TRANSPATH® [11], and we are using Colored Petri Nets [7] (among others) as the specification language. The corresponding simulation tool is CPN Tools [8]. TRANSPATH® is a database that is accessible by means of the usual methods, i.e., web interface, text files, XML (using its own XML format), etc. Coloured Petri nets are a formal representation for distributed discrete systems that allow concurrent events to be represented.

In [15], data is extracted from the TRANSPATH® database and introduced in the *CPN Tools* application manually. This implies that the user/biologist who is going to perform the simulation must manually query the database to extract the list of reactions involved in the signal transduction *pathway* to be studied. With the extracted data the corresponding Petri net must be built in the simulation tool manually.

TECHNIQUES AND TOOLS

Model-Driven Engineering (MDE) is a Software Engineering field that over the years has represented software artifacts as models in order to increase productivity, quality and to reduce costs in the software development process. Nowadays, there is increasing interest in this field, as demonstrated by the OMG guidelines that support this trend with the Model-Driven Architecture (MDA [12]) approach. Model-Driven Software Development (MDSD) has evolved from Model-Driven Engineering. MDSD not only involves design and code generation tasks, but also traceability capabilities, meta-modeling features, model persistence and model interchange tasks, etc. To address these tasks, operations between models, transformations, and queries over these models are relevant problems that must be resolved. In the MDA context, they are resolved from the open standards point of view. The Meta Object Facility (MOF) standard [14] provides support for the meta-modeling capabilities. The Query/Views/Transformations (QVT [13]) standard describes how to provide support to queries and transformations. In contrast to other new languages, QVT uses the pre-existent *Object Constraint Language* (OCL) language to perform queries over software artifacts.
MOMENT [3] is a tool that provides support to the OMG standards giving capabilities to transform models. The tool uses both an industrial modeling front-end and an algebraic back-end for the execution of the transformation and query tasks. The algebraic background runs in the high performance rewriting system Maude [4]. The industrial modeling environment used by MOMENT is the Eclipse Modeling Framework (EMF). EMF [2] can be considered as an implementation of the MOF standard and can import software artifacts from several heterogeneous data sources: UML models, XML Schemas, etc. The tool offers an implementation of the QVT-Relations language as well as the OCL language. QVT-Relations is a declarative transformations language that provides

implicit traceability capabilities. For this language, MOMENT gives wide support for unidirectional transformations. Moreover, the tools provides full support to the query operators of the OCL language.

A MDSD APPROACH IN BIOLOGICAL DATA MIGRATION

In the initial work on the study of the TLR4 signal transduction *pathway*, data migration from the source database to the simulation tool (to represent this information as a coloured Petri net) was done manually.

The solution to the data migration problem is described as follows by means of model transformation techniques using the model-driven software development guides. This implies the following tasks: (a) development of the source domain data model (TRANSPATH®), (b) development of the target domain data model (*CPN Tools*), (c) definition of the transformation rules between the source domain and the target domain by means of the transformations language, (d) implementation of the pre-processing mechanism to obtain the instances of the source model from the original data; and finally, (e) definition of the post-processing tasks that persist the transformed data in the final file format.

Architecture and overview of the tool

The data migration process is performed in three steps: (1) recovering and pre-processing of the input data, (2) execution of the transformation by means of the transformations engine and (3) post-processing and persistence of the result data. In a MDSD approach, using a transformation engine implies that the source and the target models of the transformation must be developed in the first place to be able to establish the mappings between the two domains.

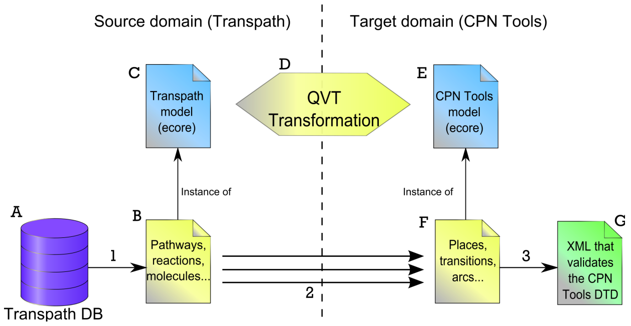
Source domain (Transpath)
Target domain (CPN Tools)
C
Transpath model (ecore)
D
QVT Transformation
E
CPN Tools model (ecore)
Instance of
Instance of
A
B
F
G
1
Pathways, reactions, molecules...
2
Places, transitions, arcs...
3
XML that validates the CPN Tools DTD
Transpath DB

FIG. 1: Architecture of the tool.

Figure 1 shows the architecture of the tool. It represents the three steps that are needed to perform the data migration. First (1), the data is extracted from the TRANSPATH® database (A), and the corresponding XMI instance (B) of the TRANSPATH® Ecore model (C) is built.

The second step (2) is the most important and complex one of the transformation process. It is performed by means of the MOMENT tool and its transformation engine. It executes the transformation from the TRANSPATH® domain (reactions, molecules, etc.) to the *CPN Tools* domain (places, transitions, arcs, etc.). After the definition of the transformation rules (D) between the source domain (C) and the target domain (E), the transformation is executed over the data recovered from the database (B) obtaining the needed information in the *CPN Tools* domain (F).

Finally, the third step (3) in the data migration process is again a trivial process in which the EMF data is stored in an XML file readable from the *CPN Tools* application (G).

Transformation process

The transformation rules that can convert data from the source domain to the target domain have been defined in a declarative way. These rules express the mappings established by biologists between the data extracted from the TRANSPATH® database and the concepts available in the *CPN Tools* application. Table I shows the simplified mappings between both the source and the target domain. The rules that define the direct relationships between the two domains have been expressed in QVT-Relations. In this language, a transformation is a set of *relations* established between the domains that participate in that transformations that must hold for the transformation to be successful [13].

Transpath	**CPN Tools**
Network	Cpnet
Pathway	Globbox – Page
Molecule (complex)	Product
Molecule (simple)	Enumerated
Reaction	Trans
Molecule (reactant)	Place – Arc (from Place to Trans)
Molecule (product)	Place – Arc (from Trans to Place)

TABLE I: Mappings between the source and the target domain.

CONCLUSIONS AND FUTURE WORK

This work has presented a case study where the interoperability problem between applications is addressed using a model-driven approach. The situation where several data sources and simulation tools co-exist and must share heterogeneous data is very common in the bioinformatics field. In this situation, the easy representation of biological data using models allows us to deal with these problems more efficiently and more elegantly than the traditional (manual) approaches.

This work presents the following advantages over the traditional approaches: (1) It allows some tasks that were previously done by hand to be automated. (2) This approach produces more modular tools, making the transformation mechanism independent from the data persistence format, improving the extensibility and maintainability of these tools. (3) Biologists do not need to know technical details about the migration process, which increases their productivity. (4) It also takes advantage of model transformation technologies. Using models to represent the data to be transformed permits the data structure to be more clearly represented making its manipulation more intuitive since it deals with high-level concepts. (5) Traceability capabilities are provided implicitly. These capabilities help to locate invalid information in the data sources. Finally, (6) using languages such as QVT-Relations offers the advantage of expressing the mappings between the source and the target domains in a declarative way. This way of representing the relationships between the two domains is more expressive than the traditional and imperative approaches.

With our case-study we presented the first steps in using model-driven techniques in the live science. Further research are focussed in different goals. First, using models to represent biological data allows us to take advantage of the new *frameworks* such as *GMF*[1] or *MS DSL Tools*. These tools use models to automatically generate visual metaphors. Second, the research done in Model-Driven Engineering can provide a rich background not only for data transformation between two different domains, but also for other tasks such as heterogeneous data integration.

ACKNOWLEDGEMENTS

We would like to thank Claudia Täubner, Dr. Silke Eckstein and Prof. Hans-Dieter Ehrich from the Institut für Informationssysteme (Technische Universität Braunschweig) for its collaboration and support during this work. We would also like to thank P. Queralt for his valuable knowledge on the use of the MOMENT-QVT tool.

[1] Gmf. `http://www.eclipse.org/gmf/`.

[2] EMF. `http://www.eclipse.org/emf/`.

[3] A. Boronat, J. Iborra, J. Ángel Carsí, I. Ramos, and A. Gómez. Del método formal a la aplicación industrial en gestión de modelos: Maude aplicado a eclipse modeling framework. *Revista IEEE América Latina*, September 2005.

[4] M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martí-Oliet, J. Meseguer, and J. F. Quesada. Maude: specification and programming in rewriting logic. *Theor. Comput. Sci.*, 285(2):187–243, 2002.

[5] K. Czarnecki and U. W. Eisenecker. *Generative programming: methods, tools, and applications.* ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 2000.

[6] S. R. S. A. e. a. Emmett, S. Towards 2020 science. Technical report, Microsoft Corporation, 2006. `http://research.microsoft.com/towards2020science/downloads/T2020S_ReportA4.pdf`.

[7] K. Jensen. *Coloured Petri Nets - Basic Concepts, Analysis Methods and Practical Use.* Springer, Berlin, 2nd edition, 1997.

[8] K. Jensen, L. Kristensen, and L. Wells. Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems. *Int. J. on Software Tools for Technology Transfer (STTT)*, Sp. Sec. CPN 04/05, 2007.

[9] G. Joshi-Tope, M. Gillespie, I. Vastrik, P. D'Eustachio, E. Schmidt, B. de Bono, B. Jassal, G. R. Gopinath, G. R. Wu, L. Matthews, S. Lewis, E. Birney, and L. Stein. Reactome: a knowledgebase of biological pathways. *Nucleic Acids Research*, 33(suppl_1):D428–432, 2005.

[10] M. Kanehisa, S. Goto, M. Hattori, K. Aoki-Kinoshita, M. Itoh, S. Kawashima, T. Katayama, M. Araki, and M. Hirakawa. From genomics to chemical genomics: new developments in KEGG. *Nucleic Acids Research*, 34(suppl_1), 2006.

[11] M. Krull, S. Pistor, N. Voss, A. Kel, I. Reuter, D. Kronenberg, H. Michael, K. Schwarzer, A. Potapov, C. Choi, O. Kel-Margoulis, and E. Wingender. TRANSPATH(R): an information resource for storing and visualizing signaling pathways and their pathological aberrations. *Nucleic Acids Research*, 34(suppl_1):D546–551, 2006.

[12] Object Management Group. MDA Guide Version 1.0.1. 2003. `http://www.omg.org/docs/omg/03-06-01.pdf`.

[13] Object Management Group. MOF 2.0 QVT final adopted specification (ptc/05-11-01). 2005. `http://www.omg.org/cgi-bin/doc?ptc/2005-11-01`.

[14] Object Management Group. Meta Object Facility (MOF) 2.0 Core Specification (ptc/06-01-01), 2006. `http://www.omg.org/cgi-bin/doc?formal/2006-01-01`.

[15] C. Taubner, B. Mathiak, A. Kupfer, N. Fleischer, and S. Eckstein. Modelling and simulation of the TLR4 pathway with coloured petri nets. *Engineering in Medicine and Biology Society, 2006. EMBS '06. 28th Annual International Conference of the IEEE*, pages 2009–2012, August 2006.